(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 June 2006 (29.06.2006)

PCT

(10) International Publication Number
WO 2006/066613 A1

(51) International Patent Classification:
G06F 17/30 (2006.01) *H04L 29/08* (2006.01)

(21) International Application Number:
PCT/EP2004/014716

(22) International Filing Date:
24 December 2004 (24.12.2004)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **TELECOM ITALIA S.P.A.** [IT/IT]; Piazza degli Affari, 2, I-20123 Milano (IT).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **BRUNO, Giorgio** [IT/IT]; Telecom Italia S.p.A., Via G. Reiss Romoli, 274, I-10148 Torino (IT). **MAMINO, Davide** [IT/IT]; Telecom Italia, S.p.A., Via G. Reiss Romoli, 274, I-10148 Torino (IT).

(74) Agents: **MARKOVINA, Paolo** et al.; Pirelli & C. S.p.A., Viale Sarca, 222, I-20126 Milan (IT).

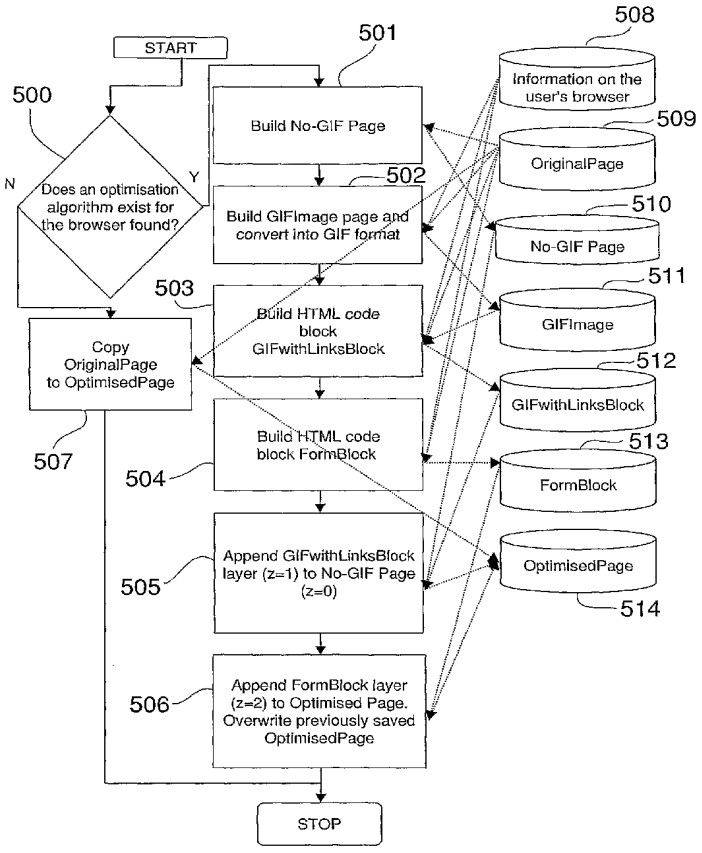
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

[Continued on next page]

(54) **Title:** METHOD OF OPTIMISING WEB PAGE ACCESS IN WIRELESS NETWORKS



(57) Abstract: The invention provides a method of optimising web page access and speeding up web page download through high-latency communications networks, such as mobile communications networks. When a user requests a web page, such page is retrieved from a web server and the image and non-image portions of the original web page are separated. Then, there are prepared (501, 502) an image-free web page including the non-image portions of the original page and having the images replaced by correspondingly sized and positioned placeholders, and an image page in which the non-image portions are made transparent and the original images are grouped into a single composite image, while keeping their positions and their sizes. The image-free page and the image page are superimposed (505, 506) to form an optimised web page, having the same appearance as the original one, which is downloaded to the user.

WO 2006/066613 A1

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHOD OF OPTIMISING WEB PAGE ACCESS IN WIRELESS NETWORKS

Field of the invention

The present invention refers to wireless communication networks, and more particularly it concerns a method of optimising access to and speeding up download of web pages in said networks.

Background of the Invention

Conventionally, domestic access to the Internet has taken place through the PSTN (Packet Switched Telephone Network), by using analogue modems. Such access is characterised by a rather limited latency between the page request and the page presentation on the user's equipment.

With such kind of access, download time of a web page depends, in a first approximation, only on the ratio between the page size (in bytes) and the channel throughput. Thus, most web pages currently available have been designed so as to ensure limited download times through a reduction of their overall sizes.

With the increasing use of wireless techniques, such as those conforming to GPRS (General Packet Radio Service), EDGE (Enhanced Data rates for GSM and IS-136 Evolution) and UMTS (Universal Mobile Telecommunication System) standards, for access to the Internet, the criterion of reducing the overall page sizes is no longer adequate, due to the very high latency time that characterises mobile wireless networks.

In a high latency network, the download time of a web page depends not only on the page size, but also on the number of objects referenced therein. In fact, a HTTP (Hyper Text Transfer Protocol) negotiation is necessary for downloading each object, and such negotiation needs a minimum time corresponding to the round trip time (RTT) of the network, which time is the sum of the network latencies in both directions.

Consequently, for a same nominal throughput of a PSTN and a wireless link (e.g. a GPRS link), access to and download of a web page through a GPRS link is much slower than through a PSTN modem.

A number of products, generally designed as Performance Enhancing Proxies (PEP), intended to decrease web page download time in high-latency networks, have become commercially available and are described in the literature. The general principles of the PEPs and of their application can be found in document RFC 3135 of The Internet Society, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations", by J. Border et al., which document is available at the site

http://www.ietf.org/rfc3135.txt.

Essentially, two classes of PEPs exist: client-server PEPs, demanding installation of a software on the client device, and client-free PEPs, which do not have such requirement.

An example of client-free PEP is disclosed in WO 03/15330 A, which teaches a parallelisation of a number of HTTP requests and, consequently, of the objects being downloaded. A data compression is further performed before forwarding the images to the user.

The client-server architectures are generally based on replacing the standard HTTP protocol by more performant protocols, and thus they are intrinsically more efficient.

Examples are disclosed in WO 01/63420 A, which teaches a system where use of predictive requests and a predictive server is made, or in US 2003/197725 A and US 6.704.024 B, which disclose systems where web pages and other visual contents are rasterised and displayed on the client device as bitmap images.

US 2001/003823 A discloses a system, based on a client-server architecture, for downloading a web page in a manner suitable for display on a television screen. This document discloses a web page conversion aimed, inter alia, at reducing the latency time. To this end, such system separates the text and image portions of an original page by downloading first the text portion of the page with any image replaced by a corresponding placeholder; then the images are retrieved and downloaded, in order to fill the placeholders. Applicants remark that, due to the use of a client-server architecture, this system has the general drawbacks inherent in this type of architecture. Moreover, downloading the text and the images in subsequent phases reduces only the apparent time of the page download, since this procedure is based on the assumption that the latency is due to the time needed by the server to provide the images themselves. The actual latency time, related to the number of HTTP negotiations, is not however reduced by such systems, since the images are individually downloaded. Moreover displaying first the text only, and then the whole page with the images, can be uncomfortable for the user.

Summary of the Invention

The Applicant has observed that the approach outlined by prior-art client-free Performance Enhancing Proxies find a limitation in the fact that known solutions are scarcely effective, as they are inherently limited by the HTTP protocols. Moreover, the degree of parallelisation of HTTP requests, cannot exceed certain limits, in order to

avoid saturation of the available upstream bandwidth.

On the other hand, the Applicant observes that PEP systems based on client-server architectures have however the drawback that they need a specific client software, which is difficult to install and to maintain.

In view of the outlined state of the art and related problems, drawbacks and limitations, the Applicant has tackled the problem of how to provide a system which is implemented by a client-free proxying server (in short, "proxy"), which is able to reduce the download time of web pages and which displays at once the whole page to the user.

According to the invention, there is provided a method in which, after separation of the image and non-image portions of a requested original web page and replacement of the images by correspondingly sized and located placeholders, at least some of the non-animated images present in the original web page are combined into a single composite image, an optimised web page is created by superimposing the composite image onto an image-free web page comprising the non-image portions of the original web page and the placeholders, and said optimised page is downloaded to a requesting user.

The creation of the optimised page comprises the steps of:

- building and temporarily storing the image-free web page;
- building and temporarily storing an image page containing the composite image, with the component images located in their original positions, and a transparent area in place of the non-image portion;
- building a first list with all links and buttons contained in the original web page, associating said first list with the composite image, and temporarily storing the first list associated with the composite image;
- building and temporarily storing a second list with all forms contained in the original web page;
- superimposing said first list associated with the composite image onto said image-free web page, and superimposing said second list onto the image-free web page having the first list associated with the composite image superimposed thereon.

Advantageously, the superimposition is obtained by means of a layering technique.

The invention also provides a computer program product loadable in the memory of at least one computer and comprising software code portions for performing the steps of the method of the invention when the product is run on a computer. As used

herein, reference to such a computer program product is intended to be equivalent to reference to a computer-readable medium containing instructions for controlling a computer system to coordinate the performance of the method of the invention. Reference to "at least one" computer is obviously intended to highlight the possibility for the arrangement of the invention to be implemented in a de-centralized fashion.

The invention provides also an apparatus for performing the method. The apparatus is essentially a client-free proxy, which can be configured either as an explicit proxy or as a user-transparent proxy.

Grouping several images of the original page into a single image significantly reduces the number of objects referenced in the page and hence the number of HTTP negotiations, and thus actually results in a significant reduction of the download time without need of employing special client software.

Brief description of the drawings

Further objects, characteristics and advantages of the invention will become apparent from the following description of preferred embodiments, given by way of non-limiting example and illustrated in the accompanying drawings, in which:

- Figs. 1 to 3 show three examples of architectures where the invention is applied;
- Fig. 4 is a general flow chart of the operation of the invention;
- Fig. 5 is a flow chart of the preparation of an optimised page; and
- Figs. 6A, 6B are HTML codes of a standard page and of the corresponding optimised page.

Description of the preferred embodiments

Three typical architectures in which the invention can be applied are disclosed with reference to Figs. 1 to 3. In such Figures, like elements are denoted by like reference numerals, beginning with digit 1 or 2 or 3, respectively.

Fig. 1 shows the application of the invention to the optimisation of the web surfing by a user. In such a situation the invention is used to speed up the access to web pages present on any public web server connected to the Internet.

In this architecture, a computer 100 has access to a mobile wireless network (GPRS/EDGE/UMTS) 103 through a mobile terminal 101 communicating with a network base station, schematised by antenna 102.

Wireless network 103 is connected to the Internet 105 through a high-throughput link 104. Through wireless network 103 and the Internet 105 the users have access to all public servers hosting the contents downloadable by the users. The drawing shows by way of example a single web server 106, connected to the Internet 105 through a

link 107.

A web page processing unit 108, implementing a page conversion according to the method of the invention, is also connected to the Internet 105 through a link 109.

Processing unit 108, referred to hereinafter as "page slicer", is essentially a client-free performance enhancement proxy exploiting the standard HTTP protocol. It attains a reduction in the latency time through a reduction of the number of images referenced in a page, and hence of the HTTP negotiations, obtained by grouping at least part of such images into a single, bigger image. Through such conversion, page slicer 108 builds and sends to the user an optimised page whose graphical aspect and functionality are the same as in the original page.

In this application page slicer 108 is to be configured as an explicit proxy, and the users are to insert the page slicer address into the web browser settings.

Fig. 2 shows the application of the invention to the optimisation of a web site. In such a situation the invention is used to speed up the access to web pages inside the network of a content provider, denoted by reference numeral 210.

As before, computer 200 has access to GPRS/EDGE/UMTS network 203 through a mobile terminal 201 communicating with a base station 202 of network 203, which is then connected to the Internet 205 through high-throughput link 204.

Content provider network 210 is connected to the Internet 205 through a high-throughput link 211 and an edge router 212. Web server 206 and page slicer 208 are located in content provider network 210 and are connected together by a link 213 with suitable throughput. Another link 214 with suitable throughput connects page slicer 208 to edge router 212.

In this architecture, page slicer 208 is transparent for the user, who therefore is not to set his/her browser so that the latter includes the page slicer address.

Fig. 3 shows the application of the invention to the optimisation of the navigation by the mobile network operator. Like in the architecture shown in Fig. 1, the invention is used to speed up the access to web pages on any public web server, like server 306, connected to the Internet. Yet, in this architecture, page slicer 308 is transparent for the user and all HTTP traffic of GPRS/EDGE/UMTS network 303 is redirected towards page slicer 308 by the control units in the wireless network. In this architecture, page slicer 308 is located between GPRS/EDGE/UMTS network 303 and the Internet 305, and is connected thereto by means of high-throughput links 315, 316, respectively.

The operations performed by page slicer 108 (or 208, 308) will now be described in detail with reference to Figs. 4 and 5.

Fig. 4 is the general flow chart of the operation.

The first step after the start of the operation is the usual request by a user (computer 100, 200 or 300 / mobile terminal 101, 201 or 301, depending on the architecture) of the web page of interest through its browser (step 400). The request arrives at page slicer 108 (208, 308), which stores the URL (Universal Resource Location) of the requested page (step 401) in a suitable memory area 406 for subsequent use.

Then page slicer 108 (208, 308) gathers and stores some information about the user's browser (steps 402, 407), i.e. it performs an identification of the client/browser pair.

Such identification is aimed at foreseeing how the original page would be rendered onto the browser, for the optimised page, which will be created according to the invention, to have exactly the same rendering. The manner in which the information about the user's browser is gathered will be discussed in detail below.

Then, at steps 403, page slicer 108 (208, 308) requests the original page (hereinafter and in the flow charts referred to as "OriginalPage") to server 106 (206, 306) and stores it in a suitable memory area 408. At step 404, the page slicer converts OriginalPage into an optimised web page (hereinafter "OptimisedPage"), which has a structure optimising access and download time through wireless network 103 (203, 303) and has the same appearance as the original page.

At step 405, page slicer 108 (208, 308) sends OptimisedPage to the user's browser, and at the same time it saves a copy thereof (as shown at 409) in a cache memory, in association with the URL and the browser characteristics, for use in case of subsequent requests.

The page slicer also implements a fallback mechanism for the case in which the client/browser pair is not recognised. In such case the conversion of step 404 is disabled to avoid supplying the user with a wrongly formatted page, and the user is supplied with the original page.

The client/browser identification also allows using the optimisation technique in connection with mobile terminals and PDAs (Personal Digital Assistants) equipped with non-standard HTML (Hyper Text Mark-up Language) browsers, which convert the format of a page according to proprietary modalities in order, for instance, to adapt the page to the display size. If the identification reveals that the browser is one such non-standard browser, and page slicer 108 (208, 308) has the information about such browser, the page will be transformed correspondingly. Otherwise, the fallback

mechanism provides for sending the original page to the user.

Turning back to the acquisition of information on the browser, the most important information items are the following ones:

- type of device being used (e.g. personal computer, PDA, Smart Phone...);
- type, release and language of the operating system;
- type, release and language of the browser;
- resolution and colour depth of the display;
- size of the viewport of the display.

To get such information, after the browser has requested OriginalPage and the page slicer 108 (208, 308) has stored the URL thereof, the page slicer sends to the browser a page containing a particular piece of JavaScript™ code collecting all of the information requested and inserts the above URL into said page, through a proper JavaScript™ variable.

Subsequently, the JavaScript™ code, after having read and stored the parameters requested, makes the browser request again OriginalPage, and the parameters are passed to the page slicer appending them to the URL of OriginalPage according to the conventional technique of the variable-value pairs.

Then, when a URL relevant to a request, with the appended parameters, arrives at the page slicer, the latter gets OriginalPage from the server, converts the page depending on the parameters and sends the converted page back to the browser.

Turning now to Fig. 5, OptimisedPage is built from OriginalPage, by using the information gathered about the client/browser pair, so as to join all non-animated images contained in a page into a single composite image. In the described embodiment, GIF (Graphical Interface Format) image format has been used, although other image formats are possible. Animated images are not processed and are still to be individually requested.

The first step 500 is a check about the existence of an optimisation algorithm for the particular browser.

If that algorithm does not exist, this means that the client/browser pair has not been identified and the fallback mechanism is implemented: OriginalPage is copied into OptimisedPage (step 507) and is downloaded to the user. The caching is performed also in this case, as shown at 514.

If the algorithm exists, page slicer 108 (208, 308) downloads the HTML code of OriginalPage from web server 106 (206, 306), performs a parsing of the code to identify all non-animated GIF images referenced therein and replaces all references by

a reference to a same transparent GIF image of 1x1 pixel (steps 501, 509). Replacement of each image is performed by maintaining the same size as the original image. If that size is not contained in the HTML code, the physical size of the image is determined and inserted into the HTML code. In this way, the graphical proportions of the page are maintained, and a transparent area or blank (placeholder) of the same size as each image is left on the page. The page so built will be referred to as "No-GIF page". The No-GIF page is stored in a temporary storage area 510, in the page slicer memory or on a disk.

At step 502, the page slicer performs the memory rendering of OriginalPage (read from memory area 509), by taking into account the client/browser pair characteristics previously gathered. Such characteristics are read from storage area 508. The page slicer eliminates all elements except the non-animated images from the page, by making such elements transparent. The result of step 502 is a single image having the same size as the browser viewport and containing only the GIF images of OriginalPage in their proper positions, whereas the remaining page portion (i.e. the text, Macromedia Flash™ content, etc.) is transparent. This image will be referred to as "GIFImage". At the same time, the page slicer converts GIFImage into GIF format, thereby creating a suitable optimised palette that is saved, as shown at 511.

Should JPEG (Joint Picture Experts Group) or PNG (Portable Network Graphics) images be contained in the page, they will be converted into GIF format so that the transparent placeholders can be built.

At step 503, a list of all links and buttons contained in OriginalPage is built, said list including the target URL and the physical position of each said link and button. Such list is associated with GIFImage in an imagemap, and an HTML code block containing said imagemap is built. This block, referred to as "GIFwithLinksBlock", is also stored in a temporary storage area in the page slicer memory or on a disk, as shown at 512.

Then, at step 504, the page slicer builds a list of all forms present in OriginalPage, together with their physical positions on the page. A further HTML code block, the "FormBlock", is built containing all said forms. FormBlock is stored in a temporary storage area in the page slicer memory or on a disk, as shown at 513.

Then the optimised page is built by the following sequence of operations.

1. OptimisedPage is initialised by creating a copy of No-GIF page; this copy forms a base layer having a co-ordinate Z = 0;
2. a layer, delimited by tags <DIV> as requested by the HTML rules and positioned

with absolute co-ordinates, is appended to this initial OptimisedPage and HTML code block "GIFwithLinksBlock " is inserted into said layer; the layer is associated with co-ordinate Z = 1 (that is, the co-ordinate of No-GIF page increased by 1), so that the layer is superimposed to the No-GIF page (step 505); the page so built is saved in storage area 514.

3. a further layer, also delimited by tags <DIV> and positioned with absolute co-ordinates, is appended to the OptimisedPage obtained by the previous step, and HTML code block "FormBlock " is inserted into said layer; the layer is associated with a co-ordinate Z = 2, so that the layer is superimposed to the previous layers (step 506). The page so built is saved at 514.

OptimisedPage is thus ready for being forwarded to the browser and displayed to the user.

An example of HTML code of OptimisedPage is shown in Fig. 6B. The different layers mentioned above are clearly apparent in the body of the HTML code.

Thanks to the way in which it has been built, OptimisedPage is aesthetically and functionally identical to OriginalPage, even if it has a different HTML code (compare Figs. 6A, 6B).

OptimisedPage contains a lower number of objects, as all GIF images have been combined into and replaced by a single composite image. Moreover, since most of the objects referenced inside web pages are typically GIF images, the reduction of the objects present in a page to be downloaded is actually significant. Since, as said above, the download time in high latency networks is strongly dependent on said number, a significant reduction of the download time of the converted page is achieved. The reduction of the objects present in a page entails a corresponding reduction of the traffic due to HTTP negotiations and thus a more advantageous exploitation of the network resources is also obtained. Moreover, building of the optimised page is independent of the browser installed on the user equipment.

Note also that, as clearly apparent from the flow charts of Figs. 4 and 5, a caching mechanism has been introduced to save the already optimised pages. In this manner, when a page already processed in the past is requested again, such page can be simply read from the cache without need of reprocessing it. This is important in view of the fact that the method is computationally complex.

The reduction of the download time can further be enhanced through a compression of the text (HTML codes, JavaScript™, CSS (Cascading Style Sheet) ...) which is to be effected by the standard compression methods for the browser to be

able to automatically perform decompression.

Similarly, also GIFImage could be compressed to reduce its size: for instance, it is possible to reduce the number of colours, or to reduce the amount of details while keeping unchanged the resolution.

It is evident that the above description has been given by way of non-limiting example and that changes and modifications are possible without departing from the scope of the invention.

For instance, even if the invention has been disclosed with particular reference to mobile communication networks, it can be applied also for optimising access through satellite links, which have features comparable to those of mobile communications links.

Moreover, even if it has been assumed hereinbefore that all GIF images in a page are joined into a single image, an alternative solution could be joining groups of GIF images physically close or adjacent to one another, thereby forming a number of composite images. The composite images can then be inserted in a layer in place of the original ones, like the single image discussed above. This alternative solution entails a smaller reduction in the download time, but it reduces implementation complexity.

Should the original page already comprise multiple layers, another alternative solution could be joining the images separately for each layer. In such situation, the caching mechanism could be performed at the layer level and thus could be more effectively exploited in case different layers have different expiry times: upon request by the browser of layers containing objects already downloaded and having an expiry time not yet elapsed, the optimised layer can be read from the cache, without need of processing again the whole page.

Patent Claims

1. A method of optimising web page access and download in wireless communication networks (103; 203; 303), the method comprising a processing of a requested original web page in order to separate image and non-image portions of said original web page and to replace the images by correspondingly sized and located placeholders, characterised in that said processing is performed in a proxying server (108; 208; 308) independent of the user and comprises the steps of:

- combining at least some of the images present in said original web page into a single composite image;
- creating an optimised web page by superimposing the composite image onto an image-free web page comprising the non-image portion of the original web page and the placeholders; and
- downloading said optimised page to a requesting user.

2. The method as claimed in claim 1, characterised in that it further comprises, before the creation of the optimised web page, the steps of:

- storing (406), upon reception (401) of a user's request for a web page, the Universal Resource Location (URL) of the requested web page; and
- gathering and storing (402, 407) information about the user's browser.

3. The method as claimed in claim 1 or 2, characterised in that it further comprises the step of saving the optimised web page (409, 514) in association with said URL of the original web page and the information about the user's browser.

4. The method as claimed in claim 2 o 3, characterised in that said step of gathering information on the user's browser includes sending a page containing a piece of JavaScript™ code by said proxying server (108; 208; 308) to said browser.

5. The method as claimed in any preceding claim, characterised in that said creation of the optimised web page comprises the steps of:

- building (501) and temporarily storing (510) said image-free web page;
- building (502) and temporarily storing (511) an image page containing said composite image, with the said at least some images located in their original positions, and transparent areas in place of the non-image portions of the original web page;
- building (503) a first list with all links and buttons contained in the original web page, associating (503) said first list with the composite image, and temporarily storing (512) the first list associated with the composite image;

- building (504) and temporarily storing (513) a second list with all forms contained in the original web page;
- superimposing (505) said first list associated with the composite image onto said image-free page, and superimposing (506) said second list onto the image-free web page having the first list associated with the composite image superimposed thereon.

6. The method as claimed in claim 5, characterised in that it comprises a preliminary check on the availability of an optimisation algorithm suitable for the user's browser, and in that the optimised web page is created through said building and superimposing steps (501 - 506) if such an algorithm is available, and is created by copying the original web page in case such algorithm is not available.

7. The method as claimed in claim 5 or 6, characterised in that said image page is converted into GIF format.

8. The method as claimed in any of claims 5 to 7, characterised in that any non-animated image in a format other than the GIF format is converted into GIF format for the replacement by placeholders and the grouping into said composite image.

9. The method as claimed in any of claims 5 to 8, characterised in that said original web page includes a Hyper Text Mark-up Language (HTML) code, and said step of building an image-free web page (501) includes:

- parsing said code to identify all non-animated GIF images referenced therein;
- replacing the references to said images by a reference to a same transparent GIF image with 1x1 pixel size.

10. A method as claimed in claim 9 when referred back to claim 5, characterised in that said step of associating (503) said first list with the composite image includes building a first HTML code block containing an imagemap providing for said association.

11. A method as claimed in claim 9 when referred back to claim 5, characterised in that said step of building (504) said second list includes building a second HTML code block containing said list.

12. A method as claimed in claim 10 or 11, characterised in that said steps of superimposing (505, 506) include:

- building, with HTML data of the image-free web page, a first layer having a first value of a co-ordinate Z;

- appending said first HTML code block at the end of the HTML data of the image-free web page, as a second layer having a co-ordinate Z increased by 1 with respect to said first value; and
- appending said second HTML code block at the end of said first HTML code block as a second layer having a co-ordinate Z increased by 2 with respect to said first value.

13. The method as claimed in any preceding claim, characterised in that said image page is compressed before being introduced into said optimised page.

14. The method as claimed in any preceding claim, characterised in that all GIF images in an original web page are combined into a single composite image

15. The method as claimed in any of claims 1 to 13, characterised in that multiple groups of GIF images in an original web page are combined into respective composite images, and said images are superimposed onto corresponding placeholders.

16. The method as claimed in claim 15, characterised in that each group comprises closely spaced or adjacent images.

17. The method as claimed in claim 15, characterised in that the original web page is organised in layers, and each group comprises the images of a layer.

18. The method as claimed in any preceding claim, characterised in that said wireless communications network is a mobile communications network exhibiting a high latency between a web page request and the page presentation on a user equipment (100, 101; 200, 201; 300, 301).

19. The method as claimed in any of claims 1 to 17, characterised in that said wireless communications network is a satellite communications network exhibiting a high latency between a web page request and the page presentation on a user equipment (100, 101; 200, 201; 300, 301).

20. An apparatus for optimising web page access and download in wireless communications networks (103; 203; 303), the apparatus comprising a processing unit (108; 208; 308) arranged to receive a web page request from a user equipment (100, 101; 200, 201; 300, 301), to download the original web page from a server (106; 206; 306) hosting it and to process said page so as to separate image and non-image portions of the page and to replace the images by correspondingly sized and located placeholders, characterised in that said processing unit (108; 208; 308) acts as a performance enhancement proxying server independent of the user equipment (100, 101; 200, 201; 300, 301), and is arranged to convert the original web page into an

optimised web page in which at least part of the images in the original web page are grouped into a single composite image and the composite image is superimposed on an image-free web page comprising the non-image portions of the original web page and the placeholders, and to download said optimised web page to the user equipment (100, 101; 200, 201; 300, 301).

21. The apparatus as claimed in claim 20, characterised in that said processing unit (108) is configured as an explicit proxying server, whose address is to included into the settings of a user's browser.

22. The apparatus as claimed in claim 20, characterised in that said processing unit (208) is configured as a transparent proxying server within a network (210) of a content provider.

23. The apparatus as claimed in claim 20, characterised in that said processing unit (308) is configured as a transparent proxying server towards which all Internet-related traffic of said wireless communications network is redirected.

24. The apparatus as claimed in any of claims 20 to 23, characterised in that it further comprises first memory units (406-409, 508, 509) storing the Universal Resource Location (URL) of the original web page, information about the equipment (100, 101; 200, 201; 300, 301) of a requesting user, the original web page and the optimised page associated with said URL of the original page and the information about the user's equipment (100, 101; 200, 201; 300, 301).

25. The apparatus as claimed in any of claims 20 to 24, characterised in that it further comprises second memory units (510, 511, 512, 513) temporarily storing:

- said image-free web page;
- an image page, containing said composite image with the said at least some images located in their original positions and transparent areas in place of the non-image portions of the original web page and having associated thereto a first list with all links and buttons contained in the original web page; and
- a second list with all forms contained in the original web page.

26. The apparatus as claimed in claim 25, characterised in that said further memory units (510, 511, 512, 513) are arranged to store HTML code blocks representative of said image-free web page, of said image page associated with the first list and of said second list by associating said code blocks with increasing consecutive values of a co-ordinate Z, and in that said processing unit (108; 208; 308) is arranged to insert said code blocks into successive layers of said optimised web page, wherein the code block representative of said image-free web page forms a

bottom layer and the code block representative of said second list forms a top layer.

27. The apparatus as claimed in any of claims 20 to 26, characterised in that said wireless communications network is a mobile communications network (103; 203; 303).

28. The apparatus as claimed in any of claims 20 to 26, characterised in that said wireless communications network is a satellite communications network.

29. A communication network including at least one web page processing unit configured for accelerating access to web pages present on web servers connected to the Internet, the network characterized in that it includes an apparatus according to any of claims 20 to 28.

30. A computer program product loadable in the memory of at least one computer and including software code portions for performing the method of any of claims 1 to 19.

31. Performance enhancing service for optimising web page access and download in wireless communication networks (10; 203; 303), by processing an original web page requested by a user equipment (100, 101; 200, 201; 300, 301) for generating an optimised web page which is downloaded to said user equipment, characterised in that said service comprises:

- providing a link to a performance enhancing proxying server, independent of the user equipment, for performing said processing of said original web page;
- causing said performance enhancing proxying server to separate image and non-image portions of the original web page and to replace the images by correspondingly sized and located placeholders, and to convert the original web page into an optimised web page in which at least part of the images in the original web page are grouped into a single composite image and the composite image is superimposed on an image-free web page comprising the non-image portions of the original web page and the placeholders;
- causing said optimised web page to be downloaded to the user equipment.

32. The service as claimed in claim 31, characterised in that said performance enhancing proxying server is configured as an explicit proxying server (108), whose address is to included into the settings of a user's browser.

33. The service as claimed in claim 31, characterised in that said performance enhancing proxying server is configured as a transparent proxying server (208) within a network (210) of a content provider.

34. The service as claimed in claim 31, characterised in that said performance

enhancing proxying server is configured as a transparent proxying server (308) towards which all Internet-related traffic of said wireless communications network is redirected.

5

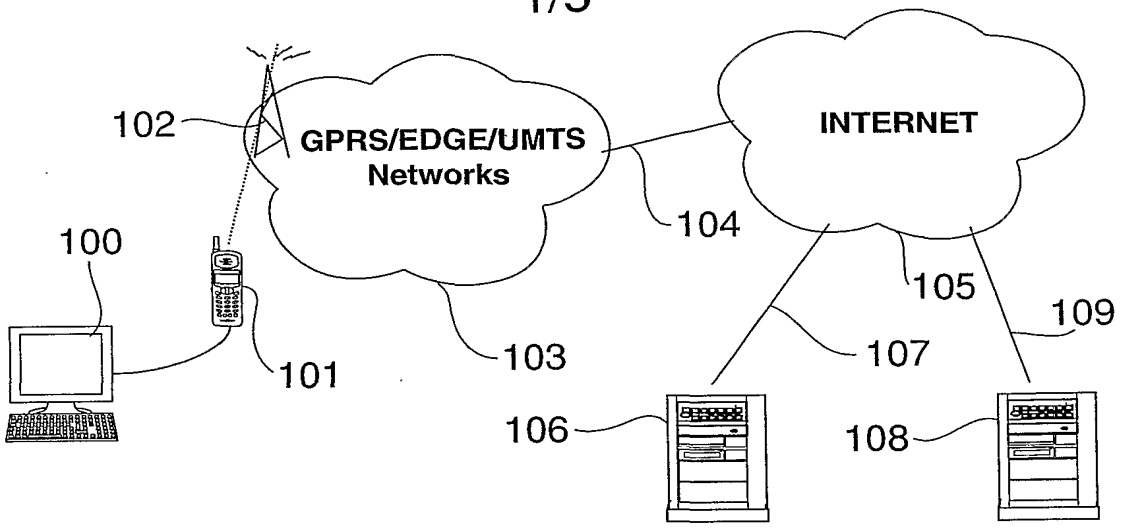

102
GPRS/EDGE/UMTS
Networks
INTERNET
100
104
105
109
103
107
101
106
108

Fig. 1

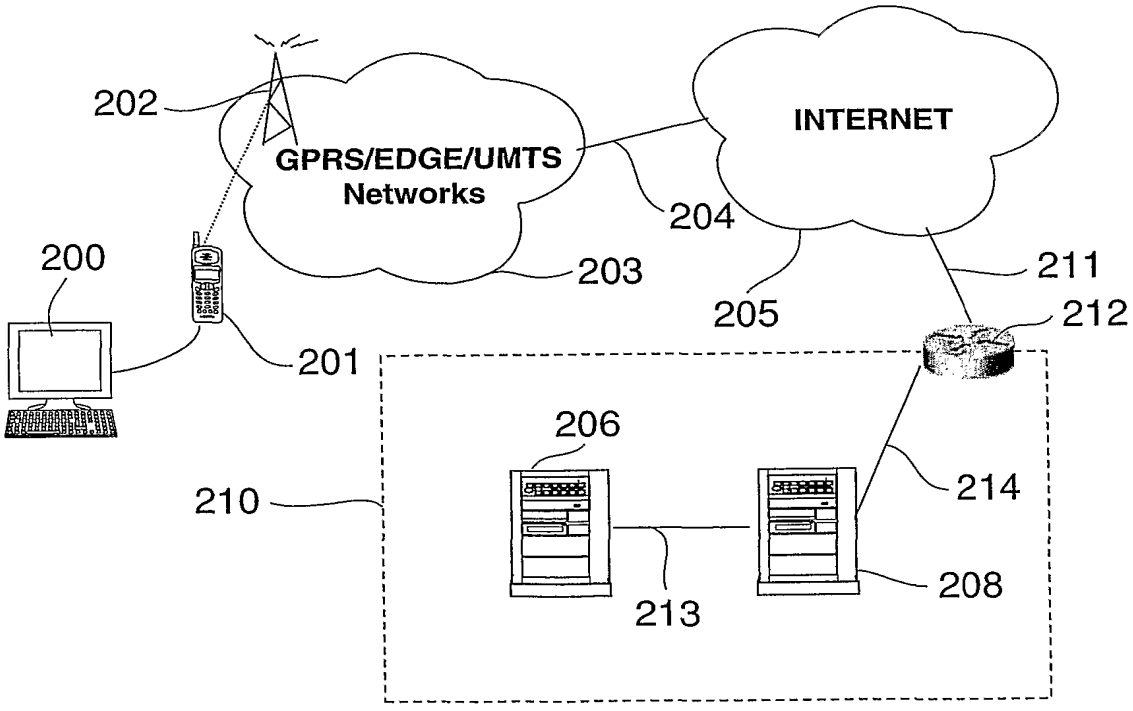

202
GPRS/EDGE/UMTS
Networks
INTERNET
204
200
203
211
205
212
201
206
210
214
213
208

Fig. 2

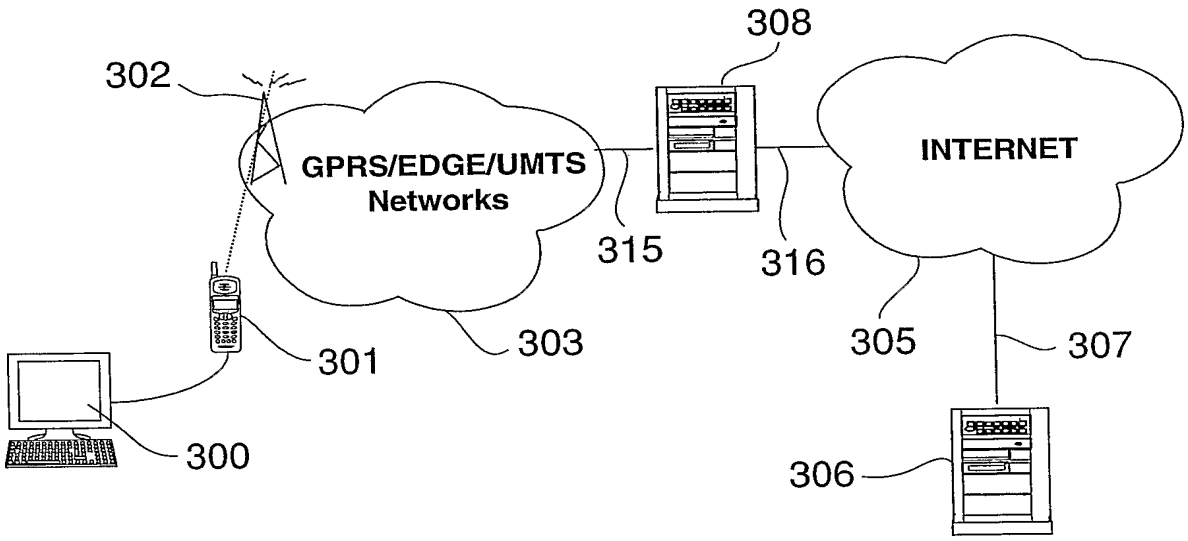

308
302
GPRS/EDGE/UMTS
Networks
INTERNET
315
316
305
307
303
301
300
306

Fig. 3

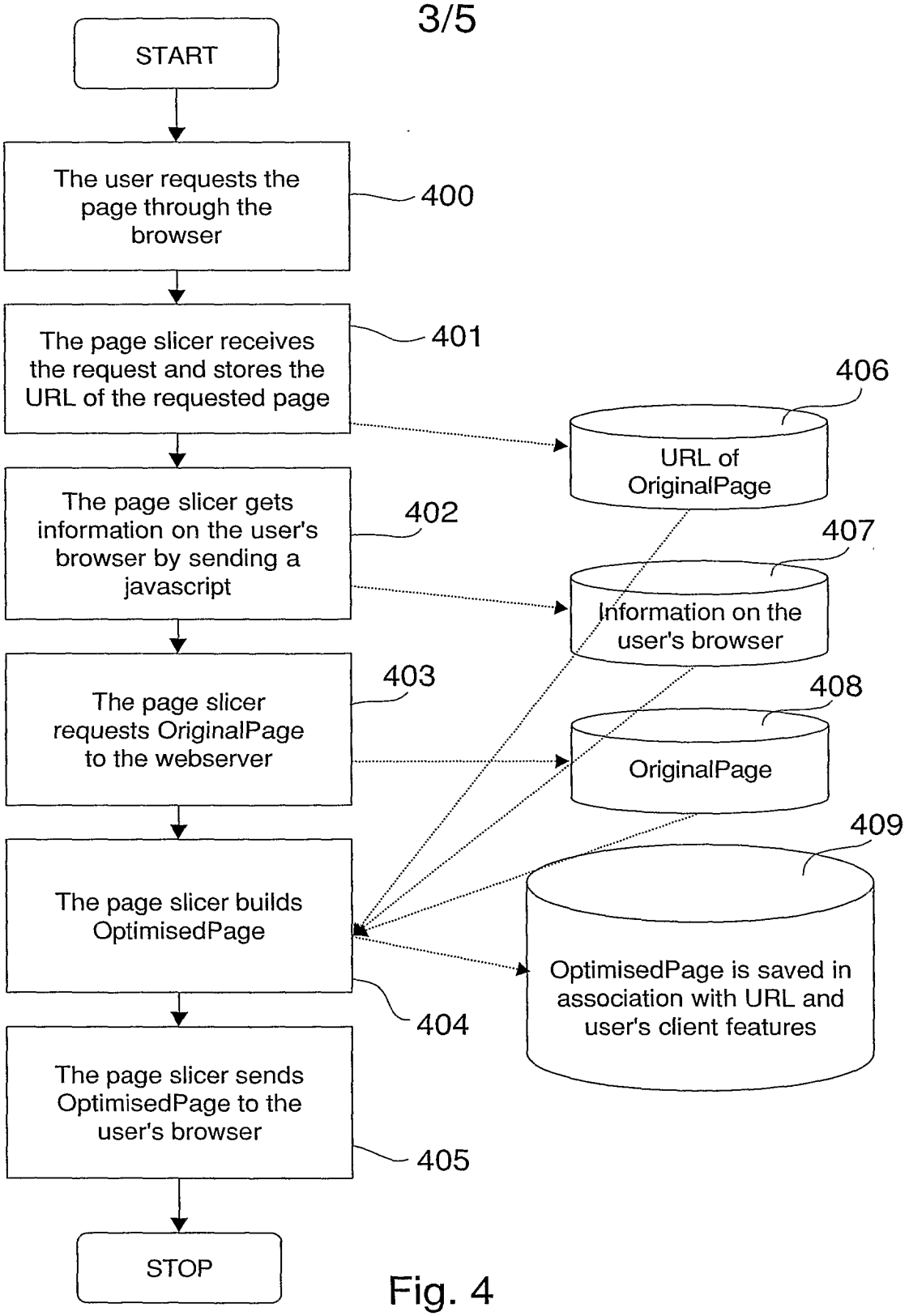

START
The user requests the page through the browser
400
The page slicer receives the request and stores the URL of the requested page
401
The page slicer gets information on the user's browser by sending a javascript
402
The page slicer requests OriginalPage to the webserver
403
The page slicer builds OptimisedPage
404
The page slicer sends OptimisedPage to the user's browser
405
STOP
406
URL of OriginalPage
407
Information on the user's browser
408
OriginalPage
409
OptimisedPage is saved in association with URL and user's client features

Fig. 4

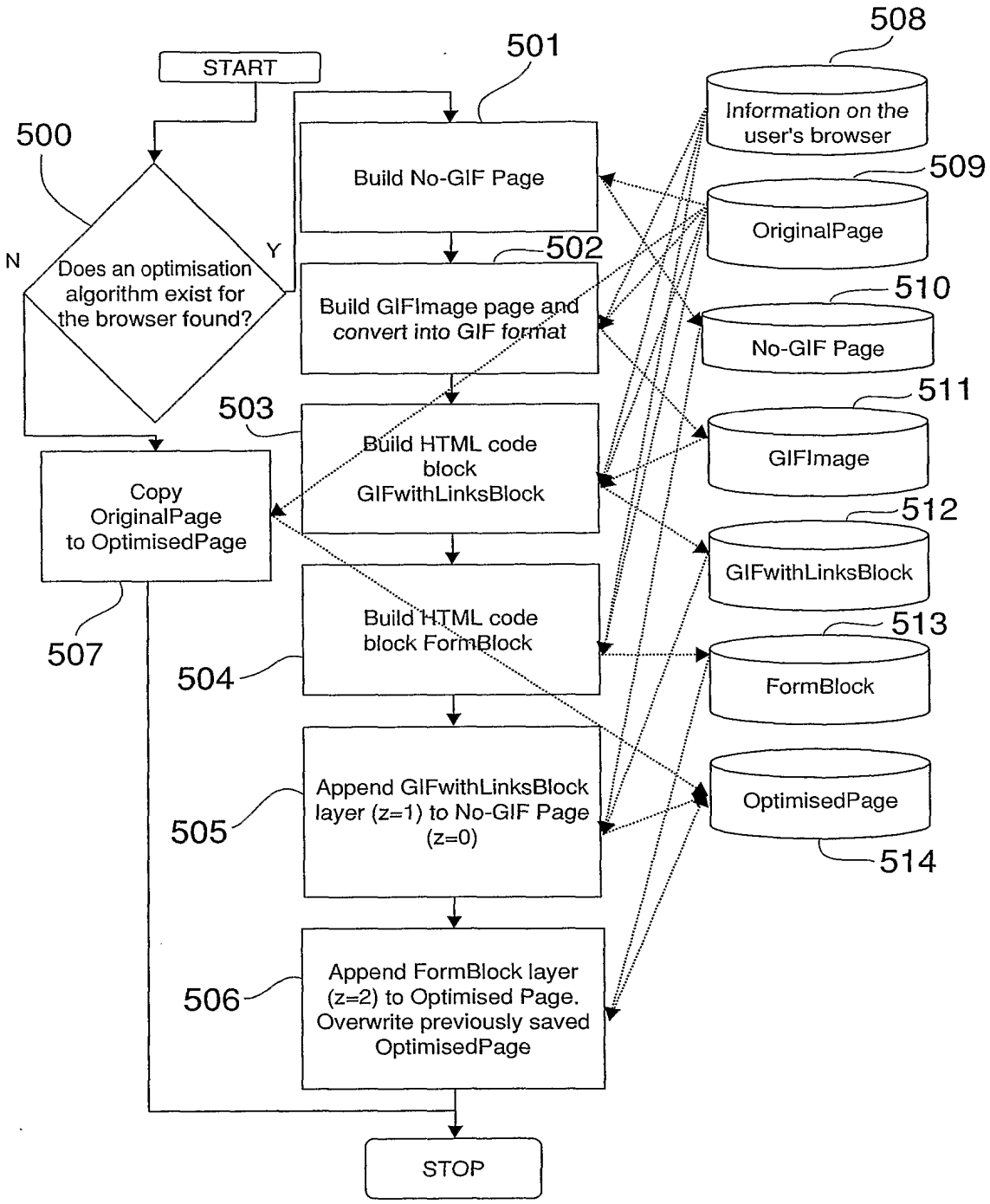

START
500
Does an optimisation algorithm exist for the browser found?
N
Y
501
Build No-GIF Page
502
Build GIFImage page and convert into GIF format
503
Build HTML code block GIFwithLinksBlock
504
Build HTML code block FormBlock
505
Append GIFwithLinksBlock layer (z=1) to No-GIF Page (z=0)
506
Append FormBlock layer (z=2) to Optimised Page. Overwrite previously saved OptimisedPage
507
Copy OriginalPage to OptimisedPage
STOP
508
Information on the user's browser
509
OriginalPage
510
No-GIF Page
511
GIFImage
512
GIFwithLinksBlock
513
FormBlock
OptimisedPage
514

Fig. 5

```
<html>
<head>
</head>

<body>
/* The whole of the HTML code of OriginalPage
is contained here */
</body>

</html>
```

Fig. 6A

```
<html>
<head>
</head>

<body>
/* The first part of the HTML code of
OptimisedPage is contained here */

<divname="GIFwithLinksBlock" style="z-index: 1;
position: absolute;">
/* The whole of the HTML code of GIFwithLinksBlock
is contained here */
</div>

<divname="FormBlock" style="z-index: 2; position:
absolute;">
/* The whole of the HTML code of FormBlock is
contained here */
</div>
</body>

</html>
```

Fig. 6B

INTERNATIONAL SEARCH REPORT

International Application No
PCT/EP2004/014716

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F17/30 H04L29/08

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2001/003823 A1 (MIGHDOLL LEE S ET AL) 14 June 2001 (2001-06-14) the whole document	1-34
A	WO 2004/097645 A (TELECOM ITALIA S.P.A; BRUNO, GIORGIO; MAMINO, DAVIDE) 11 November 2004 (2004-11-11) claims	1,20, 29-31
A	WO 03/015330 A (FLASH NETWORKS LTD; BAR-YAACOV, ASSAF; AVIRAN, SHARON) 20 February 2003 (2003-02-20) claims; figures	1,20, 29-31
A	US 2004/177148 A1 (TSIMELZON MARK) 9 September 2004 (2004-09-09) paragraph '0069! - paragraph '0090!	1-3

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
21 June 2005	30/06/2005

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Herry, T

Form PCT/ISA/210 (second sheet) (January 2004)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/EP2004/014716

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2001003823 A1	14-06-2001	US 5918013 A	29-06-1999
		US 5896444 A	20-04-1999
		US 6034689 A	07-03-2000
		AU 3375197 A	05-01-1998
		EP 0811939 A2	10-12-1997
		JP 10228437 A	25-08-1998
		KR 274738 B1	15-12-2000
		US 6647421 B1	11-11-2003
		US 6505232 B1	07-01-2003
		WO 9746943 A1	11-12-1997
		US 6023268 A	08-02-2000
		US 6230319 B1	08-05-2001
		US 6141693 A	31-10-2000
		US 2003014499 A1	16-01-2003
		US 6330606 B1	11-12-2001
		US 6311207 B1	30-10-2001
		US 6332157 B1	18-12-2001
		US 6259442 B1	10-07-2001
		US 6308222 B1	23-10-2001
		US 5940074 A	17-08-1999
		US 6073168 A	06-06-2000
		US 5935207 A	10-08-1999
		US 5996022 A	30-11-1999
		US 5974461 A	26-10-1999
		US 2002013812 A1	31-01-2002
		AU 3227597 A	05-01-1998
		EP 0812096 A2	10-12-1997
		JP 10155039 A	09-06-1998
		KR 274739 B1	15-12-2000
		WO 9747124 A1	11-12-1997
		US 6308221 B1	23-10-2001
		US 2002048354 A1	25-04-2002
		US 6278773 B1	21-08-2001
		US 6584506 B1	24-06-2003
		US 6587886 B1	01-07-2003
		AU 3139197 A	05-01-1998
		EP 0811940 A2	10-12-1997
		JP 10171842 A	26-06-1998
		KR 274135 B1	15-12-2000
		WO 9747143 A2	11-12-1997
		US 6133913 A	17-10-2000
		US 6496205 B1	17-12-2002
		US 6473099 B1	29-10-2002
		US 6008836 A	28-12-1999
		US 6005563 A	21-12-1999
		US 5945991 A	31-08-1999
		US 2002054069 A1	09-05-2002
		US 2002021308 A1	21-02-2002
WO 2004097645 A	11-11-2004	WO 2004097645 A1	11-11-2004
		AU 2003233206 A1	23-11-2004
WO 03015330 A	20-02-2003	EP 1415232 A2	06-05-2004
		WO 03015330 A2	20-02-2003
		US 2004177152 A1	09-09-2004
US 2004177148 A1	09-09-2004	US 6763388 B1	13-07-2004
		AU 6761900 A	05-03-2001

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/EP2004/014716

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2004177148 A1		AU 6762000 A	05-03-2001
		AU 6762100 A	05-03-2001
		WO 0111488 A2	15-02-2001
		WO 0111502 A2	15-02-2001
		WO 0111503 A2	15-02-2001